

CLAIMS

1. A method for analyzing a binary RDO file structure, extracting all relevant
5 data needed to reproduce content thereof, and generating an output in a
selected format, comprising the steps of:

reading and analyzing said binary RDO file;

extracting data contained within said RDO file describing an
arrangement of pages in a final document; and

10 generating an output by placing one or more bitmap files for each page
onto an output page and adding optional text messages for header, footer,
and page number.

2. The method of Claim 1, said reading and analyzing step further
15 comprising:

decoding said binary RDO file internal structure;

parsing said binary RDO file; and

transferring said parsed binary RDO file into a data structure
representation in a memory.

20 3. The method of Claim 1, said extracting step further comprising:

collecting data for each page in said RDO binary file, where said data
are scattered throughout said RDO binary file, and where some data are
page-invariant and that apply to an entire document embodied in said RDO
25 binary file.

4. The method of Claim 3, wherein said page-invariant data comprise any of header and footer messages, their location, or font selection or margin specifications.

5 5. The method of Claim 1, wherein said bitmap file is a TIFF format file.

6. The method of Claim 1, further comprising the step of:

storing said output in a memory when all pages have been processed.

10 7. The method of Claim 1, wherein said selected format is a PDF format.

8. The method of Claim 1, wherein said bitmap file is a PostScript file and wherein an external, commercially available Postscript-to-PDF converter is invoked to merge these pages into an output PDF.

15

9. An apparatus for analyzing a binary RDO file structure, extracting all relevant data needed to reproduce content thereof, and generating an output in a selected format, comprising:

a read module for reading and analyzing said binary RDO file;

20

an understand module for extracting data contained within said RDO file describing an arrangement of pages in a final document; and

a reproduce module for generating an output by placing a bitmap file for each page onto an output page and adding optional text messages for header, footer, and page number.

25

10. The apparatus of Claim 9, said read module further comprising:

a decoder for decoding said binary RDO file internal structure;
a parser for parsing said binary RDO file; and
a memory for receiving a data structure representation of said parsed
binary RDO file.

5

11. The apparatus of Claim 9, said understand module further comprising:

a mechanism for collecting data for each page in said RDO binary file,
where said data are scattered throughout said RDO binary file, and where
some data are page-invariant and that apply to an entire document embodied
in said RDO binary file.

10

12. The apparatus of Claim 11, wherein said page-invariant data comprise
any of header and footer messages, their location, or font selection.

15

13. The apparatus of Claim 9, wherein said bitmap file is a TIFF format file.

14. The apparatus of Claim 9, further comprising:

a memory for storing said output when all pages have been processed.

20

15. The apparatus of Claim 9, wherein said selected format is a PDF format.

16. The apparatus of Claim 9, wherein said bitmap file is a PostScript file.

17. The apparatus of Claim 16, further comprising:

25

an external, commercially available Postscript-to-PDF converter for
merging said bitmap file for each of said pages into an output PDF.

18. The apparatus of Claim 9, wherein said output comprises an internal representation of any of the following items once all data have been gathered from said RDO file:

5 for each page a list of images on a page; optional header, footer, and page number strings; location of text items; and fonts, font attributes, and sizes to be used;

 for each image image dimensions; orientation and offset and alignment information; and information about layering of multiple images on top of one
10 another;

 for said RDO document a list of page images and page numbers; a list of sections; font selection for header, footer and page number; and margins;

 for each section a list of page images and page numbers.

15 19. A method for analyzing a binary RDO file structure, extracting all relevant data needed to reproduce content thereof, and generating an output in a selected format, comprising the steps of:

 reading and analyzing said binary RDO file;

 extracting data contained within said RDO file describing an
20 arrangement of pages in a final document; and

 generating an output by placing one or more bitmap files for each page onto an output page and adding optional text messages for header, footer, and page number decoding said binary RDO file internal structure;

 parsing said binary RDO file into a tree data structure; and

25 transferring said parsed binary RDO file as said tree data structure representation to a memory.

20. The method of Claim 19, wherein said step of parsing said tree structure comprises an initialization function which reads said RDO binary into memory and a recursive parsing function.

5

21. The method of Claim 20, wherein said initialization function comprises the step of:

reading said RDO file into a buffer, wherein a first code byte is read, a size byte is read, and said parsing function is invoked.

10

22. The method of Claim 21, wherein said parsing function comprises the steps of:

reading the next code;

making a determination if said code is a leaf and, if so, said leaf data are read and stored and said process continues, wherein if said code is read as a directory, then a next size is read and, if said size read does not fit into a remaining byte size, then an error is detected and said process is aborted, otherwise remaining size is reduced by a new size and said parsing function is invoked to effect recursion, wherein upon return, a child tree is then stored, and if a remaining size is greater than zero said process is repeated, otherwise said process terminates.

15

20

23. The method of Claim 19, wherein said extracting step comprises any of:

creating a template similar to an expected subtree and then attempting to match said template against all trees in said RDO file in a recursive fashion, wherein a matching algorithm returns pointers to sought leaves of a matching

25

RDO tree, and wherein once said template has been matched, desired values can be read back from said pointers; and

looping through all trees and calling a specific handler routine based on the code of a topmost directory of each tree, wherein a handler routine then (optionally recursively) attempts to follow a certain path of subdirectories through a subtree based on a predetermined sequence of codes to read desired leaves with said data, and wherein said data are then stored in a fashion that associates different pieces with images or pages in said document.

24. The method of Claim 19, further comprising:

providing a separate job ticket file which specifies printing options that are not directly part of said document and that depend on capabilities of an output device; and

extracting information stored in said job ticket file, which information relates to features supported by a particular device or set of devices.

25. The method of Claim 24, wherein said job ticket files specifies any of:

number of copies of said document to be printed;

a range of pages of said document which are to be printed;

sides of a page of said document which are to be printed;

paper stock to be used for printing said document, wherein a paper stock may have any of the following properties: size, type, drilled or not, color, weight per unit area, stapling options, and collation;

distance by which image is to be shifted while printing;

whether a job is to be printed or to be stored in a particular file;

information that is useful for identifying a job, which information may include of: name of a document to be printed, name of a user who is sending a request to print, account, deliver to, banner message, and special instructions;

- 5 special instructions for including several sets of exception pages which describe pages which are to be printed with printer settings that are different from those defined for said document as a whole, wherein an exception page specification may have any of the following components: range of pages, paper stock to be used, sides imaged, and image shift specifications;

- 10 special instructions for inserting pages in said job from alternative sources, which instruction may comprise any of the following components: page number after which pages are to be inserted, number of pages to be inserted, and paper stock to be used; and

- 15 type of covers that must be printed for a particular job, which may specify any of the following: where a cover is required, front cover paper stock if required, back cover paper stock if required, sides to be printed for front cover if required, and sides to be printed for back cover if required.